1/18/2008

- What did we talk about last class? Have you thought about what you would like to do in this course?
- What do you know about how we represent numbers on computers?

- Before we get started talking about how floating point numbers are stored let's write a small bit of code using some numerics. In particular let's explore two different fractals.
  - ◆ The Mandelbrot set is defined as the set of points, C, in the complex plane where the following sequence stays finite: $z_{n+1}=z_n^2+C$, $z_0=C$.
  - ◆ Serpenski's triangle can be made by taking 3 points in the plane, then picking a 4th point at random. Pick one of the 3 points randomly and move halfway to it. Repeat this plotting out the new point each time.
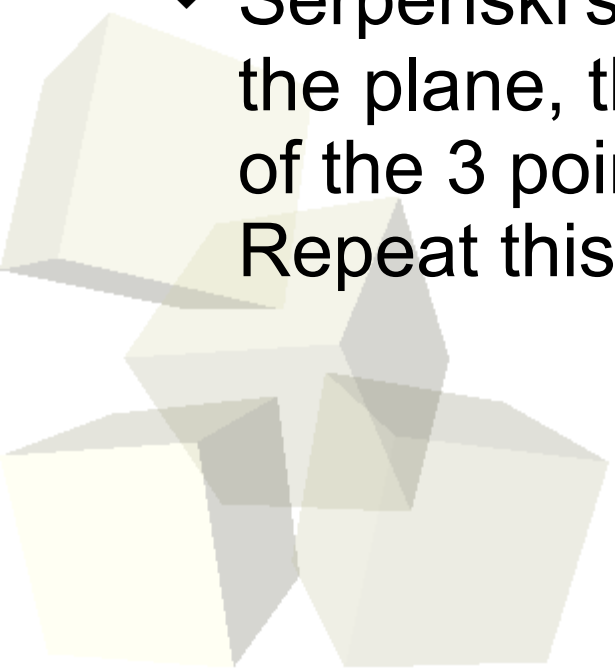
- We have Matlab installed on these machines. Why don't you go ahead and start up Matlab and we will do the rest of the day in it, even though what we are going to do could be done in C/C++.
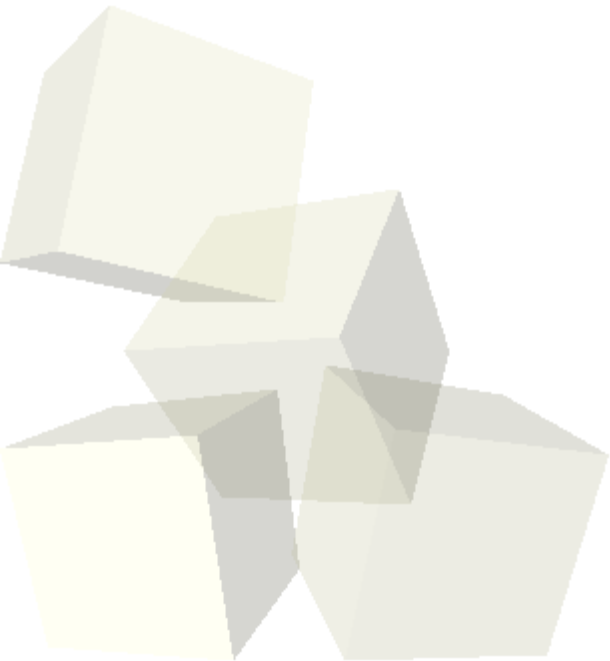
- Computers represent all numbers in binary (base 2). For fractional numbers we typically us a notation called floating point. This is very much like scientific notation using binary.
- The standard for floating point is IEEE 754. This comes in both single and double precision. Both styles have one byte for the sign followed by bits for an exponent and a fractional part.
- For single precision the exponent gets 8 bits (with a bias of 127) and the fraction/mantissa gets the remaining 23.
- Double precision gets 11 for the exponent (1023 biased) and 52 for the fraction.
- Both assume a leading 1.

- There are some aspects of the floating point representation that are interesting to note. In particular, the differences between integer and floating point representation.
  - The mantissa has a sign bit instead of using 2s complement.
  - The exponent has a bias instead of being 2s complement.
- Combined with the ordering of sign, exponent, mantissa you get much faster comparisons.
- The leading 1 makes 0 a special form. Some implementations have unnormalized forms for numbers below the range of the standard.

- The details of IEEE 754 don't matter to most people. What does matter is that these numbers are not the "real" numbers from math.
- Having limited precision means some properties, like associativity, don't always hold.
- The real key is that not all numbers can be represented perfectly. This is similar to the fact that we can't represent numbers like 1/3rd in decimal with a finite number of digits.
- Similarly, there are some types of operations that should be avoided.
  - Don't subtract two numbers that differ by an amount much smaller than their magnitude.
  - Don't divide by very small numbers.

- These rules of thumb are part of the reason many scientists will use tools like Matlab or existing libraries. Often these libraries will use special alternate formulas in situations where the "normal" one is poorly behaved.
- It is often difficult enough just to get the math right without worrying too much about how the numerics will behave.

- Matlab has a built in constant called eps which is the smallest number such that 1+eps>1. Write a loop in C/C++ that would calculate eps.
- Have a good extended weekend and remember to read chapters 4 and 5 from the Matlab book for next class.